

Notes on Linear Algebra

Edward Gyubeom Im*

January 31, 2023

Contents

1	Linear Equation	4
2	Linear System	4
3	Identity Matrix	4
4	Inverse Matrix	4
5	Solving Linear System	4
6	Linear Combination	5
7	Span	5
8	From Matrix Equation to Vector Equation	5
9	Several Perspectives about Matrix Multiplication	5
10	Linear Independence	6
11	Linear Dependence	7
12	Span and Subspace	8
13	Basis of a Subspace	8
14	Dimension of Subspace	8
15	Column Space of Matrix	8
16	Rank of Matrix	9
17	Transformation	9
18	Linear Transformation	9
19	Transformations between Vectors	9
20	Matrix of Linear Transformation	9
21	Onto and One-To-One	10
22	Least Square	10
23	Inner Product	10

*blog: alida.tistory.com, email: gyurse@gmail.com

24 Properties of Inner Product	10
25 Vector Norm	11
26 Unit Vector	11
27 Distance between Vectors in \mathbb{R}^n	11
28 Inner Product and Angle between Vectors	11
29 Orthogonal Vectors	11
30 Least Square Problem	12
31 Normal Equation	12
32 Another Derivation of Normal Equation	13
33 What If $C = A^T A$ is NOT Invertible?	13
34 Orthogonal Projection Perspective	13
35 Orthogonal and Orthonormal Sets	13
36 Orthogonal and Orthonormal Basis	13
37 Orthogonal Projection \hat{y} of y onto Line	14
38 Orthogonal Projection \hat{y} of y onto Plane	14
39 Orthogonal Projection when $y \in W$	14
40 Transformation: Orthogonal Projection	15
41 Orthogonal Projection Perspective	15
42 Gram-Schmidt Orthogonalization	15
43 Eigenvectors and Eigenvalues	16
44 Null Space	16
45 Orthogonal Complement	16
46 Characteristic Equation	17
47 Eigenspace	17
48 Diagonalization	17
49 Finding V and D	17
50 Eigendecomposition	18
51 Linear Transformation via Eigendecomposition	18
51.1 Change of Basis	18
51.2 Element-wise Scaling	18
51.3 Back to Original Basis	19
52 Linear Transformation via A^k	19
53 Geometric Multiplicity and Algebraic Multiplicity	19

54	Singular Value Decomposition	19
55	SVD as Sum of Outer Products	20
56	Another Perspective of SVD	20
57	Computing SVD	20
58	Diagonalization of Symmetric Matrices	21
59	Spectral Theorem of Symmetric Matrices	21
60	Spectral Decomposition	21
61	Positive (Semi-)Definite Matrices	21
62	Symmetric Positive Definite Matrices	21
63	Back to Computing SVD	21
64	Eigendecomposition in Machine Learning	22
65	Low Rank Approximation of a Matrix	22
66	Dimension Reducing Transformation	22
67	Derivative of multi-variable function	22
67.1	Gradient	22
67.2	Jacobian matrix	23
67.2.1	Toy example	23
67.3	Hessian matrix	23
67.4	Laplacian	24
67.5	Taylor expansion	24
68	Reference	24
69	Revision log	24

1 Linear Equation

선형방정식(Linear Equation)은 변수 x_1, \dots, x_n 이 있을 때 다음과 같이 작성할 수 있는 방정식을 의미한다.

$$a_1x_1 + a_2x_2 + \dots + a_nx_n = b \quad (1)$$

이 때, b 는 계수를 의미하고 a_1, \dots, a_n 값들은 실수 또는 복소수의 미지수를 의미한다. 위 식은 다음과 같이 간결하게 작성할 수 있다.

$$\mathbf{a}^T \mathbf{x} = b \quad (2)$$

이 때, $\mathbf{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}$ 이고 $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ 이다.

2 Linear System

선형방정식(Linear Equation)의 집합을 선형시스템(Linear System)이라고 한다. n 개의 선형방정식 $\mathbf{a}_1\mathbf{x} = b_1, \dots, \mathbf{a}_n\mathbf{x} = b_n$ 이 있는 경우 이를 다음과 같이 간결하게 선형시스템으로 표현할 수 있다.

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (3)$$

3 Identity Matrix

항등행렬(Identity Matrix)은 대각성분이 전부 1이고 나머지 성분이 전부 0인 $n \times n$ 크기의 정방행렬을 의미한다. 일반적으로 $\mathbf{I} \in \mathbb{R}^{n \times n}$ 으로 표현한다.

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3 \times 3} \quad (4)$$

항등행렬에 임의의 벡터 $\mathbf{x} \in \mathbb{R}^n$ 을 곱하면 자기 자신이 도출된다.

$$\forall \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{I}\mathbf{x} = \mathbf{x} \quad (5)$$

4 Inverse Matrix

정방행렬 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 에 대한 역행렬(Inverse Matrix) \mathbf{A}^{-1} 는 다음과 같이 정의된다.

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I} \quad (6)$$

2×2 크기의 정방행렬 $\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ 가 있을 때, 역행렬은 다음과 같이 정의된다.

$$\mathbf{A}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \quad (7)$$

3×3 크기 이상의 정방행렬도 역행렬을 구할 수 있다. 역행렬은 정방행렬이면서 Full Rank인 행렬(= non-singular, $\det \mathbf{A} \neq 0$)에만 존재한다.

5 Solving Linear System

행렬 \mathbf{A} 의 역행렬이 존재하는 경우 선형시스템은 역행렬을 사용하여 다음과 같이 풀 수 있다.

$$\begin{aligned} \mathbf{Ax} &= \mathbf{b} \\ \mathbf{A}^{-1}\mathbf{Ax} &= \mathbf{A}^{-1}\mathbf{b} \\ \mathbf{Ix} &= \mathbf{A}^{-1}\mathbf{b} \\ \mathbf{x} &= \mathbf{A}^{-1}\mathbf{b} \end{aligned} \quad (8)$$

그러나, 행렬 \mathbf{A} 의 판별식 $\det \mathbf{A} = 0$ 인 경우 역행렬이 존재하지 않게되고 위와 같이 문제를 풀 수 없다. 이런 경우 선형시스템은 해가 존재하지 않거나 무수히 많은 해가 존재한다.

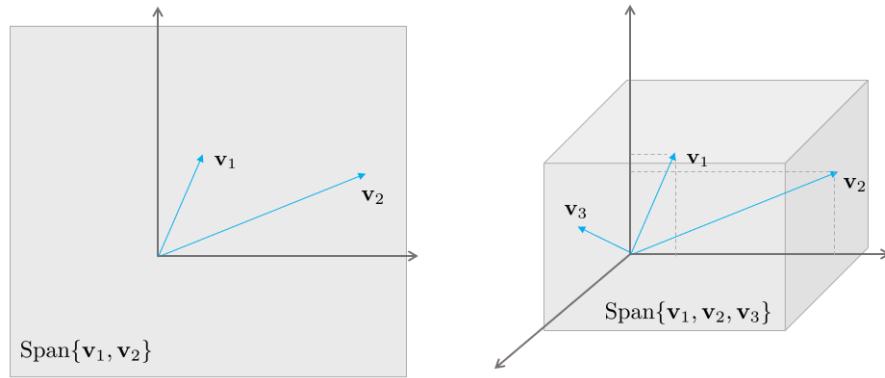
6 Linear Combination

여러 벡터 $\mathbf{v}_1, \dots, \mathbf{v}_n \in \mathbb{R}^n$ 이 있을 때 스칼라 값 c_1, \dots, c_n 에 대하여

$$c_1\mathbf{v}_1 + \dots + c_n\mathbf{v}_n \quad (9)$$

을 벡터 $\mathbf{v}_1, \dots, \mathbf{v}_n$ 의 가중치 계수 c_1, \dots, c_n 에 대한 **선형결합 (Linear Combination)**이라고 한다. 이 때 가중치 계수 c_1, \dots, c_n 는 0을 포함한 실수 값을 가진다.

7 Span



주어진 여러 벡터 $\mathbf{v}_1, \dots, \mathbf{v}_n \in \mathbb{R}^n$ 에 대해 $\text{Span}\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ 은 모든 $\mathbf{v}_1, \dots, \mathbf{v}_n$ 에 대한 선형결합의 집합을 의미한다. 즉, $\text{Span}\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ 은 다음과 같이 쓸 수 있는 모든 벡터들의 집합이다.

$$c_1\mathbf{v}_1 + \dots + c_n\mathbf{v}_n \quad (10)$$

이는 또한 $\mathbf{v}_1, \dots, \mathbf{v}_n$ 에 의해 span 된 \mathbb{R}^n 공간 상의 subset이라고도 불린다.

8 From Matrix Equation to Vector Equation

$\mathbf{Ax} = \mathbf{b}$ 와 같은 선형 시스템을 다음과 같이 열벡터 \mathbf{a}_i 를 기준으로 펼쳐보면

$$[\mathbf{a}_1 \quad \mathbf{a}_2 \quad \cdots \quad \mathbf{a}_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \mathbf{b} \quad (11)$$

로 나타낼 수 있고 이를 다시 표현하면

$$\mathbf{a}_1x_1 + \mathbf{a}_2x_2 + \cdots + \mathbf{a}_nx_n = \mathbf{b} \quad (12)$$

와 같이 **열벡터들의 선형결합으로 표현**할 수 있게 된다. 만약 \mathbf{b} 가 $\text{Span}\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ 에 포함되어 있다면 이들의 선형결합으로 표현할 수 있으므로 해가 존재한다. 따라서 $\mathbf{b} \in \text{Span}\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ 일 때 해가 존재한다.

9 Several Perspectives about Matrix Multiplication

선형시스템 $\mathbf{Ax} = \mathbf{b}$ 가 있을 때 이는 곧 \mathbf{A} 의 열벡터들의 선형결합으로 표현할 수 있다.

$$\mathbf{Ax} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \mathbf{a}_1x_1 + \mathbf{a}_2x_2 + \cdots + \mathbf{a}_nx_n = \mathbf{b} \quad (13)$$

만약 선형시스템에 전치행렬을 적용하여 $\mathbf{x}^\top \mathbf{A}^\top = \mathbf{b}^\top$ 가 되면

$$\begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix} = \mathbf{a}_1 x_1 + \mathbf{a}_2 x_2 + \cdots + \mathbf{a}_n x_n = \mathbf{b} \quad (14)$$

\mathbf{b}^\top 는 곧 \mathbf{A}^\top 의 행벡터(Row Vector)들의 선형결합으로 표현된다.

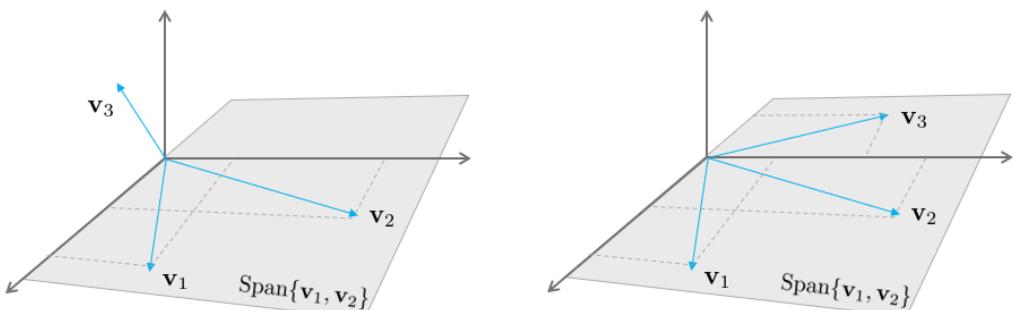
또한 두 벡터의 곱 $\mathbf{ab}^\top = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} \begin{bmatrix} b_1 & \cdots & b_n \end{bmatrix}$ 의 경우 rank1 outer product로 볼 수 있다. 즉, $[\mathbf{a} \ \mathbf{c}] \begin{bmatrix} \mathbf{b} \\ \mathbf{d} \end{bmatrix}$

의 경우 $\mathbf{ab} + \mathbf{cd}$ 와 같이 벡터곱을 스칼라 곱과 같이 생각할 수 있다.

10 Linear Independence

벡터 집합 $\mathbf{v}_1, \dots, \mathbf{v}_n \in \mathbb{R}^n$ 가 주어졌을 때, 이들 중 부분 벡터들의 집합 $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{j-1}\}$ 이 선형결합을 통해 특정 벡터 $\mathbf{v}_j, j = 1, \dots, n$ 를 표현할 수 있는지 검사한다.

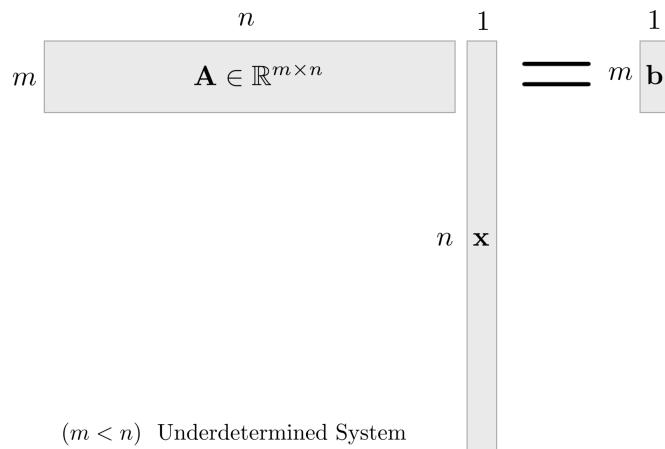
$$\mathbf{v}_j \in \text{Span}\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{j-1}\} \quad \text{for some } j = 1, \dots, n? \quad (15)$$

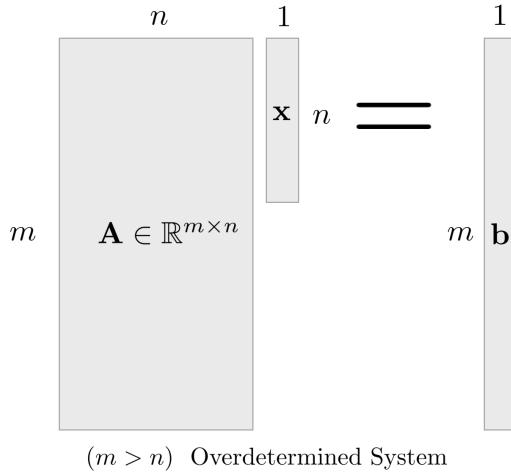


Linearly Independent
 $\mathbf{v}_3 \notin \text{Span}\{\mathbf{v}_1, \mathbf{v}_2\}$

Linearly Dependent
 $\mathbf{v}_3 \in \text{Span}\{\mathbf{v}_1, \mathbf{v}_2\}$

만약 \mathbf{v}_j 가 선형결합으로 표현이 된다면 $\mathbf{v}_1, \dots, \mathbf{v}_n$ 는 선형의존 (Linearly Dependent)이다. 만약, \mathbf{v}_j 가 표현되지 않는다면 $\mathbf{v}_1, \dots, \mathbf{v}_n$ 는 선형독립 (Linearly Independent)이다.





($m > n$) Overdetermined System

$\mathbf{Ax} = \mathbf{b}$ 와 같은 시스템에서 \mathbf{A} 행렬이 $\mathbb{R}^{m \times n}$ 이고 $m < n$ 인 경우 방정식의 개수보다 미지수의 개수가 많으므로 \mathbf{A} 의 열벡터에 의해 Span되는 공간이 항상 \mathbf{b} 의 차원보다 크게 되어 무수히 많은 해가 존재한다. 이를 Under-determined System이라고 한다. 반대로 $m > n$ 인 경우 미지수의 개수보다 방정식의 개수가 많으므로 해가 존재하지 않는데 이를 Over-determined System이라고 한다.

만약 $x_1\mathbf{v}_1 + x_2\mathbf{v}_2 + \cdots + x_n\mathbf{v}_n = \mathbf{0}$ 같은 동차(homogeneous) 선형방정식이 있다고 하면

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (16)$$

과 같은 자명해가 존재한다. 이 때, $\mathbf{v}_1, \dots, \mathbf{v}_n$ 가 선형독립이면 자명해 이외에 해는 존재하지 않는다. 하지만, $\mathbf{v}_1, \dots, \mathbf{v}_n$ 가 선형의존이면 선형시스템은 자명해 이외에 다른 해가 존재한다.

자명해 이외에 다른 해가 존재하는 선형의존(Linearily Dependent) 경우 대해서 생각해보면 예를 들어 \mathbf{A} 행렬이 다음과 같이 5개의 열을 가진 행렬이라고 했을 때

$$\mathbf{A} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \mathbf{a}_3 \ \mathbf{a}_4 \ \mathbf{a}_5] \quad (17)$$

위 열벡터(Column Vector)들 중 최소한 두 개 이상의 벡터가 선형결합되어야 동차방정식 $\mathbf{Ax} = \mathbf{0}$ 의 해를 만족할 수 있다. 예를 들어 \mathbf{a}_2x_2 성분이 0이 아닌 경우 이를 다시 영벡터로 만들기 위해서는 다른 1,3,4,5 열벡터들의 선형결합이 $-\mathbf{a}_2x_2$ 의 값을 만들어야 한다. 이는 곧 \mathbf{a}_2x_2 값을 다른 열벡터들의 선형결합으로 표현할 수 있다는 말과 동치이므로 선형의존인 경우 어떤 하나의 벡터가 다른 벡터들의 선형결합으로 표현될 수 있음을 의미한다. 이를 수식으로 표현하면 다음과 같다.

$$\begin{aligned} \mathbf{a}_jx_j &= -\mathbf{a}_1x_1 - \cdots - \mathbf{a}_{j-1}x_{j-1} \\ \mathbf{a}_j &= -\frac{x_1}{x_j}\mathbf{a}_1 - \cdots - \frac{x_{j-1}}{x_j}\mathbf{a}_{j-1} \in \text{Span}\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{j-1}\} \end{aligned} \quad (18)$$

11 Linear Dependence

행렬 \mathbf{A} 의 열벡터 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$ 이 선형의존(Linearily Dependent)인 경우 해당 열벡터들은 Span의 차원을 늘리지 않는다. 만약 $\mathbf{A} \in \mathbb{R}^{3 \times 3}$ 이고 $\mathbf{a}_3 \in \text{Span}\{\mathbf{a}_1, \mathbf{a}_2\}$ 인 경우

$$\text{Span}\{\mathbf{a}_1, \mathbf{a}_2\} = \text{Span}\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\} \quad (19)$$

만약 $\mathbf{a}_3 = d_1\mathbf{a}_1 + d_2\mathbf{a}_2$ 와 같이 선형결합으로 표현이 가능한 경우, $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ 는 다음과 같이 작성할 수 있다.

$$c_1\mathbf{a}_1 + c_2\mathbf{a}_2 + c_3\mathbf{a}_3 = (c_1 + d_1)\mathbf{a}_1 + (c_2 + d_2)\mathbf{a}_2 \quad (20)$$

12 Span and Subspace

\mathbb{R}^n 공간의 부분공간(Subspace) H는 \mathbb{R}^n 의 부분집합들의 선형결합에 대해 닫혀 있는 공간을 의미한다. 즉, 두 벡터 $\mathbf{u}_1, \mathbf{u}_2 \in H$ 일 때, 어떠한 스칼라 값 c, d 에 대하여 $c\mathbf{u}_1 + d\mathbf{u}_2 \in H$ 일 때 H를 부분공간이라고 한다.

Span $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ 으로 형성된 공간은 항상 부분공간이다. 만약 $\mathbf{u}_1 = x_1\mathbf{a}_1 + \dots + x_n\mathbf{a}_n$ 이고 $\mathbf{u}_2 = y_1\mathbf{a}_1 + \dots + y_n\mathbf{a}_n$ 일 때

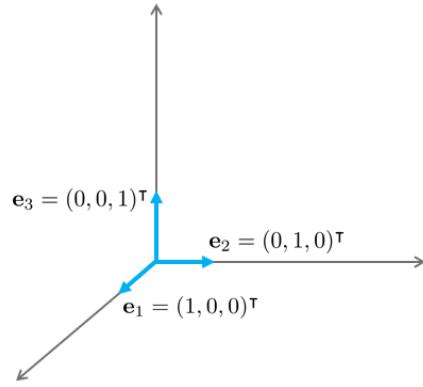
$$\begin{aligned} c\mathbf{u}_1 + d\mathbf{u}_2 &= c(x_1\mathbf{a}_1 + \dots + x_n\mathbf{a}_n) + d(y_1\mathbf{a}_1 + \dots + y_n\mathbf{a}_n) \\ &= (cx_1 + dy_1)\mathbf{a}_1 + \dots + (cx_n + dy_n)\mathbf{a}_n \end{aligned} \quad (21)$$

과 같이 선형결합으로 나타낼 수 있고 이는 임의의 값 c, d 에 대해서 닫혀 있음을 의미한다. 따라서 부분공간은 항상 Span $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ 으로 표현된다.

13 Basis of a Subspace

부분공간 H의 기저(basis)는 다음을 만족하는 벡터들의 집합을 의미한다.

1. 부분공간 H를 모두 Span할 수 있어야 한다.
2. 벡터들 간 선형독립이어야 한다.



3차원 공간 \mathbb{R}^3 의 경우 기저벡터는 3개가 존재하고 $\mathbf{e}_1 = [1 \ 0 \ 0]^\top$, $\mathbf{e}_2 = [0 \ 1 \ 0]^\top$, $\mathbf{e}_3 = [0 \ 0 \ 1]^\top$ 일 때, 이를 표준기저벡터(Standard Basis Vector)라고 한다.

14 Dimension of Subspace

하나의 부분공간 H를 표현할 수 있는 기저는 유일하지 않다. 하지만 여러개의 기저를 통해서 표현할 수 있는 부분공간의 차원(Dimension)은 유일하다. 부분공간의 차원은 기저벡터의 개수와 동일하다.

15 Column Space of Matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_{1,col} & \mathbf{a}_{i,col} & \mathbf{a}_{m,col} \\ \begin{matrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{matrix} \end{bmatrix} \in \mathbb{R}^{n \times m} \quad \text{Col } \mathbf{A} = \text{Span}\{\mathbf{a}_{1,col}, \mathbf{a}_{i,col}, \mathbf{a}_{m,col}\}$$

행렬 \mathbf{A} 의 열공간(Column Space)이란 \mathbf{A} 의 열벡터로 인해 Span된 부분공간을 의미한다. 일반적으로 Col \mathbf{A} 라고 표기한다.

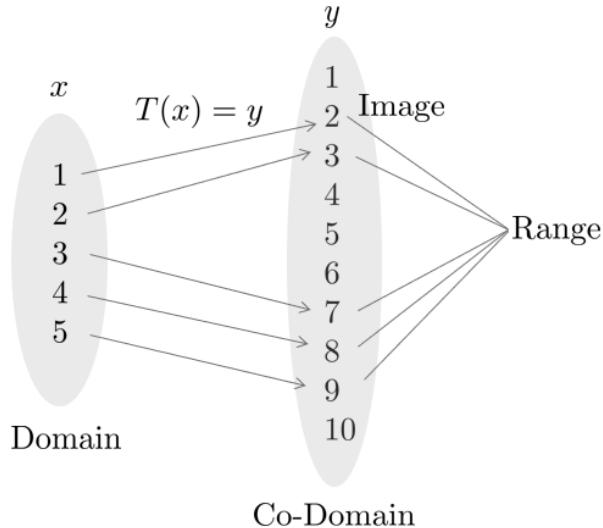
$$\text{Col } \mathbf{A} = \text{Span}\left\{\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}\right\} \quad (22)$$

16 Rank of Matrix

행렬 \mathbf{A} 의 rank란 \mathbf{A} 의 열벡터들의 차원을 의미한다.

$$\text{rank}\mathbf{A} = \dim \text{Col}\mathbf{A} \quad (23)$$

17 Transformation



변환(Transformation), 함수(Function), 매팅(Mapping) T 은 입력 x 를 출력 y 로 매팅해주는 것을 의미한다.

$$T : x \mapsto y \quad (24)$$

이 때 입력 x 에 의해 매팅되는 출력 y 는 유일하게 결정된다. **Domain**(정의역)이란 입력 x 의 모든 가능한 집합을 의미한다. **Co-Domain**(공역)이란 출력 y 의 모든 가능한 집합을 의미한다. **Image**란 주어진 입력 x 에 대해 매팅된 출력 y 를 의미한다. **Range**(치역)란 Domain내에 있는 입력 x 들에 의해 매팅된 모든 출력 y 의 집합을 의미한다.

18 Linear Transformation

변환 T 는 다음과 같은 경우에 선형변환(Linear Transformation)이라고 한다.

$$T(c\mathbf{u} + d\mathbf{v}) = cT(\mathbf{u}) + dT(\mathbf{v}) \quad (25)$$

for all \mathbf{u}, \mathbf{v} in the domain of T and for all scalars c and d .

19 Transformations between Vectors

$T : \mathbf{x} \in \mathbb{R}^n \mapsto \mathbf{y} \in \mathbb{R}^m$ 은 n 차원의 벡터를 m 차원의 벡터로 매팅하는 연산을 의미한다. 예를 들면

$$T : \mathbf{x} \in \mathbb{R}^3 \mapsto \mathbf{y} \in \mathbb{R}^2$$

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \in \mathbb{R}^3 \mapsto \mathbf{y} = T(\mathbf{x}) = \begin{bmatrix} 4 \\ 5 \end{bmatrix} \in \mathbb{R}^2 \quad (26)$$

20 Matrix of Linear Transformation

변환 $T : \mathbb{R}^n \mapsto \mathbb{R}^m$ 을 선형변환이라고 가정하면 T 는 항상 행렬과 벡터의 곱으로 표현할 수 있다. 즉,

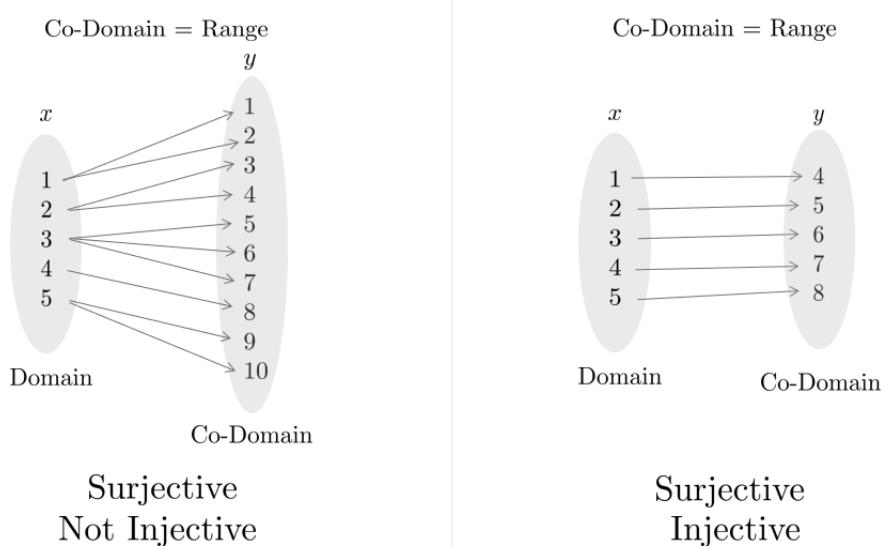
$$T(\mathbf{x}) = \mathbf{Ax} \quad \text{for all } \mathbf{x} \in \mathbb{R}^n \quad (27)$$

행렬 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 인 경우 \mathbf{A} 의 j 번째 열 \mathbf{a}_j 는 벡터 $T(\mathbf{e}_j)$ 와 같다. 이 때 \mathbf{e}_j 는 항등행렬 $\mathbf{I} \in \mathbb{R}^{n \times n}$ 의 j 번째 열벡터이다.

$$\mathbf{A} = [T(\mathbf{e}_1) \ \cdots \ T(\mathbf{e}_n)] \quad (28)$$

이러한 행렬 \mathbf{A} 를 선형변환 T 의 표준행렬(Standard Matrix)이라고 부른다.

21 Onto and One-To-One



Onto는 전사함수(Surjective)라고도 불리며 공역이 치역과 같은 경우를 의미한다. 이는 Co-Domain의 모든 원소들이 사영된 것을 의미한다.

$$\text{Surjective: Co-Domain} = \text{Range} \quad (29)$$

One-To-One은 일대일함수(Injective)라고도 불리며 정의역의 원소와 공역의 원소가 하나씩 대응되는 함수를 의미한다.

22 Least Square

최소제곱법(Least Square)은 방정식의 개수가 미지수의 개수보다 많은 Over-determined 선형시스템에서 사용하는 방법 중 하나이다. Over-determined 선형시스템 $\mathbf{Ax} = \mathbf{b}$ 의 경우 일반적으로 해가 존재하지 않는다. 이런 경우 일반적으로 $\|\mathbf{Ax} - \mathbf{b}\|^2$ 가 최소가 되는 근사해를 구할 수 있다.

23 Inner Product

벡터 $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ 에 대해 이를 각각 $n \times 1$ 의 행렬로 생각할 수 있다. 그렇다면 \mathbf{u}^\top 는 $1 \times n$ 의 행렬로 볼 수 있고 행렬곱 $\mathbf{u}^\top \mathbf{v}$ 는 1×1 의 행렬이 된다. 그리고 1×1 행렬은 스칼라값으로 표시할 수 있다.

이 때, $\mathbf{u}^\top \mathbf{v}$ 에 의해 계산된 값을 \mathbf{u}, \mathbf{v} 의 내적(Inner Product, Dot Product)라고 한다. 이는 $\mathbf{u} \cdot \mathbf{v}$ 로 표기할 수 있다.

24 Properties of Inner Product

벡터 $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbb{R}^n$ 이고 c 를 스칼라 값이라고 할 때 내적은 다음과 같은 성질을 만족한다.

1. $\mathbf{u} \cdot \mathbf{v} = \mathbf{v} \cdot \mathbf{u}$
2. $(\mathbf{u} + \mathbf{v}) \cdot \mathbf{w} = \mathbf{u} \cdot \mathbf{w} + \mathbf{v} \cdot \mathbf{w}$

3. $(c\mathbf{u}) \cdot \mathbf{v} = c(\mathbf{u} \cdot \mathbf{v}) = \mathbf{u} \cdot (c\mathbf{v})$

4. $\mathbf{u} \cdot \mathbf{u} \geq 0$ and $\mathbf{u} \cdot \mathbf{u} = 0$ iff $\mathbf{u} = 0$

위에서 2,3번 성질을 조합하면 다음과 같은 법칙을 만들 수 있다.

$$(c_1\mathbf{u}_1 + \cdots + c_n\mathbf{u}_n) \cdot \mathbf{w} = c_1(\mathbf{u}_1 \cdot \mathbf{w}) + \cdots + c_n(\mathbf{u}_n \cdot \mathbf{w}) \quad (30)$$

위를 통해 **내적이라는 연산은 선형변환이라는 것을 알 수 있다.**

25 Vector Norm

벡터 $\mathbf{v} \in \mathbb{R}^n$ 에 대해 벡터의 놈(Norm)은 0이 아닌 $\|\mathbf{v}\| = \sqrt{\mathbf{v} \cdot \mathbf{v}}$ 로 표기하며 벡터의 길이를 의미한다.

$$\|\mathbf{v}\| = \sqrt{\mathbf{v} \cdot \mathbf{v}} = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2} \quad (31)$$

2차원 벡터 $\mathbf{v} \in \mathbb{R}^2$ 가 있을 때 $\mathbf{v} = \begin{bmatrix} a \\ b \end{bmatrix}$ 라고 하면 $\|\mathbf{v}\|$ 는 원점으로부터 \mathbf{v} 좌표까지의 거리가 된다.

$$\|\mathbf{v}\| = \sqrt{a^2 + b^2} \quad (32)$$

모든 스칼라 값 c 에 대해 $c\mathbf{v}$ 의 길이는 \mathbf{v} 의 길이를 $|c|$ 배 한 것을 의미한다.

$$\|c\mathbf{v}\| = |c| \|\mathbf{v}\| \quad (33)$$

26 Unit Vector

길이가 1인 벡터를 단위벡터(Unit Vector)라고 한다. 벡터의 길이를 1로 맞추는 작업을 정규화(Normalization)라고 하는데 주어진 벡터 \mathbf{v} 가 있을 때 단위벡터 $\mathbf{u} = \frac{1}{\|\mathbf{v}\|}\mathbf{v}$ 가 된다. \mathbf{u} 벡터는 \mathbf{v} 벡터와 방향은 같지만 크기가 1인 벡터이다.

27 Distance between Vectors in \mathbb{R}^n

두 벡터 $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ 이 있을 때 두 벡터의 거리는 $\text{dist}(\mathbf{u}, \mathbf{v})$ 로 나타내며 이는 $\mathbf{u} - \mathbf{v}$ 벡터의 길이를 의미한다.

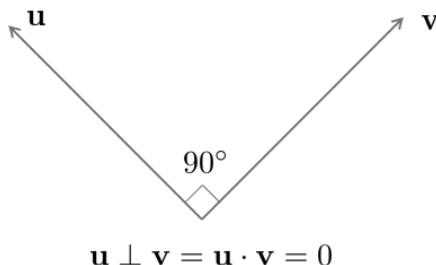
$$\text{dist}(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\| \quad (34)$$

28 Inner Product and Angle between Vectors

두 벡터 \mathbf{u}, \mathbf{v} 의 내적은 다음과 같이 놈과 각도를 통해 표현할 수 있다.

$$\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\| \|\mathbf{v}\| \cos \theta \quad (35)$$

29 Orthogonal Vectors



두 벡터 $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ 가 있을 때 둘이 수직이려면 두 벡터의 내적이 0이어야 한다.

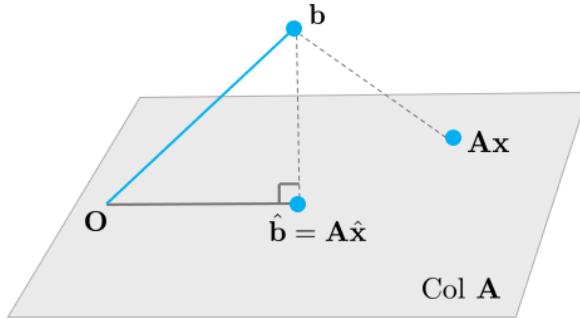
$$\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\| \|\mathbf{v}\| \cos \theta = 0 \quad (36)$$

0이 아닌 두 벡터 \mathbf{u}, \mathbf{v} 의 내적이 0이려면 $\cos \theta$ 값이 0이어야 하고 $\theta = 90^\circ$ 일 때 $\cos \theta$ 값은 0이 된다.

30 Least Square Problem

$\mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{b} \in \mathbb{R}^n, m \ll n$ 과 같이 주어진 Over-Determined 시스템 $\mathbf{Ax} = \mathbf{b}$ 가 있을 때 에러의 제곱합 $\|\mathbf{b} - \mathbf{Ax}\|$ 을 최소화하는 최적의 모델 파라미터를 찾는 것이 목적이 된다. 이 때 최소제곱법의 근사해 $\hat{\mathbf{x}}$ 는 다음과 같다.

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\mathbf{b} - \mathbf{Ax}\| \quad (37)$$



$$\mathbf{Ax} = \mathbf{b} \Rightarrow \mathbf{A}\hat{\mathbf{x}} = \hat{\mathbf{b}}$$

최소제곱법의 중요한 포인트 중 하나는 어떤 \mathbf{x} 파라미터를 선정하던지 벡터 \mathbf{Ax} 는 반드시 $\text{Col } \mathbf{A}$ 안에 위치한다는 것이다. 따라서 **최소제곱법은 Col A와 b의 거리가 최소가 되는 x를 찾는 문제가 된다.**

$\hat{\mathbf{b}} = \mathbf{A}\hat{\mathbf{x}}$ 를 만족하는 근사해 $\hat{\mathbf{x}}$ 는 $\text{Col } \mathbf{A}$ 에서 \mathbf{b} 벡터와 가장 가까운 모든 포인트들의 집합을 의미한다. 따라서 \mathbf{b} 는 다른 어떤 \mathbf{Ax} 보다도 $\hat{\mathbf{b}}$ 와 가장 가깝게 된다. 기하학적으로 이를 만족하기 위해서는 벡터 $\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}$ 가 $\text{Col } \mathbf{A}$ 와 수직이어야 한다.

$$\mathbf{b} - \mathbf{A}\hat{\mathbf{x}} \perp (x_1 \mathbf{a}_1 + x_2 \mathbf{a}_2 + \cdots + x_n \mathbf{a}_n) \text{ for any vector } \mathbf{x}. \quad (38)$$

이는 곧 다음과 동일하다.

$$\begin{aligned} (\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}) \perp \mathbf{a}_1 &\rightarrow \mathbf{a}_1^\top (\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}) \\ (\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}) \perp \mathbf{a}_2 &\rightarrow \mathbf{a}_2^\top (\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}) \\ (\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}) \perp \mathbf{a}_3 &\rightarrow \mathbf{a}_3^\top (\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}) \\ \therefore \mathbf{A}^\top (\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}) &= 0 \end{aligned} \quad (39)$$

31 Normal Equation

$\mathbf{Ax} \approx \mathbf{b}$ 를 만족하는 최소제곱법의 근사해는 다음과 같다.

$$\mathbf{A}^\top \mathbf{A}\hat{\mathbf{x}} = \mathbf{A}^\top \mathbf{b} \quad (40)$$

위 식을 **정규방정식(Normal Equation)**이라고 부른다. 이는 $\mathbf{C} = \mathbf{A}^\top \mathbf{A} \in \mathbb{R}^{n \times n}, \mathbf{d} = \mathbf{A}^\top \mathbf{b} \in \mathbb{R}^n$ 일 때 $\mathbf{Cx} = \mathbf{d}$ 와 같은 선형시스템으로 생각할 수 있다. 이 선형시스템의 해를 구하면 다음과 같다.

$$\hat{\mathbf{x}} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b} \quad (41)$$

32 Another Derivation of Normal Equation

근사해 $\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\mathbf{b} - \mathbf{Ax}\| = \arg \min_{\mathbf{x}} \|\mathbf{b} - \mathbf{Ax}\|^2$ 와 같이 제곱을 최소화하는 문제로 표현해도 동일한 문제가 된다.

$$\arg \min_{\mathbf{x}} (\mathbf{b} - \mathbf{Ax})^T (\mathbf{b} - \mathbf{Ax}) = \mathbf{b}^T \mathbf{b} - \mathbf{x}^T \mathbf{A}^T \mathbf{b} - \mathbf{b}^T \mathbf{Ax} + \mathbf{x}^T \mathbf{A}^T \mathbf{Ax} \quad (42)$$

위 식을 \mathbf{x} 에 대해서 미분하고 정리하면 다음과 같다.

$$-\mathbf{A}^T \mathbf{b} - \mathbf{A}^T \mathbf{b} + 2\mathbf{A}^T \mathbf{Ax} = 0 \Leftrightarrow \mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b} \quad (43)$$

이 때 $\mathbf{A}^T \mathbf{A}$ 가 역행렬이 존재한다면 다음과 같이 해를 구할 수 있다.

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \quad (44)$$

33 What If $\mathbf{C} = \mathbf{A}^T \mathbf{A}$ is NOT Invertible?

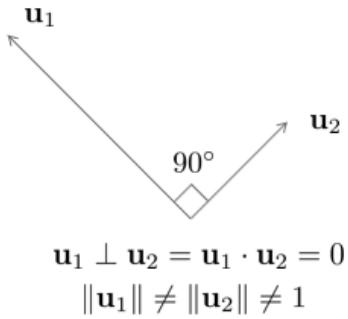
행렬 $\mathbf{C} = \mathbf{A}^T \mathbf{A}$ 의 역행렬이 존재하지 않는 경우 시스템은 해가 없거나 무수히 많은 해를 가지고 있다. 하지만 정규방정식은 항상 해를 가지고 있으므로 해가 없는 상황은 존재하지 않고 실제로는 무수히 많은 해를 가지고 있다. \mathbf{C} 가 역행렬을 구할 수 없는 경우는 오직 Col \mathbf{A} 가 선형의존일 경우에 발생한다. 하지만, 일반적으로 \mathbf{C} 는 대부분의 경우 역행렬이 존재한다.

34 Orthogonal Projection Perspective

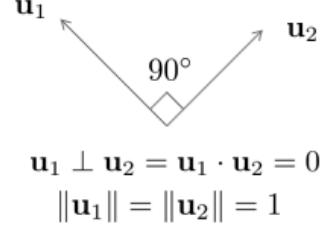
행렬 $\mathbf{C} = \mathbf{A}^T \mathbf{A}$ 가 있을 때 \mathbf{b} 점에서 Col \mathbf{A} 공간으로 프로젝션하면 다음과 같다.

$$\hat{\mathbf{b}} = f(\mathbf{b}) = \mathbf{A} \hat{\mathbf{x}} = \mathbf{A} (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \quad (45)$$

35 Orthogonal and Orthonormal Sets



Orthogonal Set



Orthonormal Set

벡터들의 집합 $\mathbf{u}_1, \dots, \mathbf{u}_n \in \mathbb{R}^n$ 가 있을 때 모든 벡터 쌍들이 $\mathbf{u}_i \cdot \mathbf{u}_j = 0$, $i \neq j$ 를 만족하면 해당 집합은 **직교(Orthogonal)**하다고 말한다.

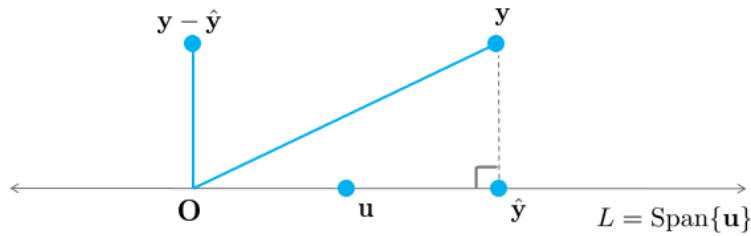
벡터들의 집합 $\mathbf{u}_1, \dots, \mathbf{u}_n \in \mathbb{R}^n$ 가 있을 때 모든 직교 집합들이 단위벡터인 경우 **정규직교(Orthonormal)**하다고 말한다.

직교벡터와 정규직교벡터의 집합은 **항상 선형독립이다**.

36 Orthogonal and Orthonormal Basis

기저벡터 $\mathbf{u}_1, \dots, \mathbf{u}_n$ 이 p차원의 부분공간 $W \in \mathbb{R}^n$ 에 있다고 할때 Gram-Schmidt 프로세스와 QR decomposition을 사용하면 직교기저벡터를 만들 수 있다. 부분공간 W 에 대해 직교기저 벡터 $\mathbf{u}_1, \dots, \mathbf{u}_n$ 이 주어져 있다고 했을 때 $\mathbf{y} \in \mathbb{R}^n$ 을 부분공간 W 위로 프로젝션시킨다.

37 Orthogonal Projection \hat{y} of y onto Line



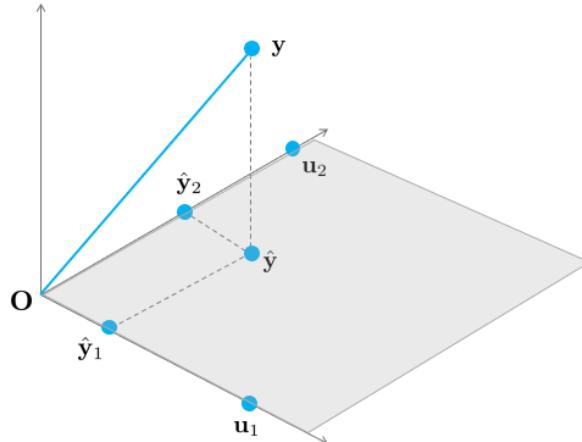
1차원 부분공간 $L = \text{Span}\{u\}$ 위로 y 를 프로젝션하여 \hat{y} 를 구하면 다음과 같다.

$$\hat{y} = \text{proj}_L y = \frac{\mathbf{y} \cdot \mathbf{u}}{\mathbf{u} \cdot \mathbf{u}} \mathbf{u} \quad (46)$$

가 된다. 만약 u 가 단위벡터이면 다음과 같다.

$$\hat{y} = \text{proj}_L y = (\mathbf{y} \cdot \mathbf{u}) \mathbf{u} \quad (47)$$

38 Orthogonal Projection \hat{y} of y onto Plane



2차원 부분공간 $W = \text{Span}\{u_1, u_2\}$ 위로 y 를 프로젝션하여 \hat{y} 를 구하면 다음과 같다.

$$\hat{y} = \text{proj}_L y = \frac{\mathbf{y} \cdot \mathbf{u}_1}{\mathbf{u}_1 \cdot \mathbf{u}_1} \mathbf{u}_1 + \frac{\mathbf{y} \cdot \mathbf{u}_2}{\mathbf{u}_2 \cdot \mathbf{u}_2} \mathbf{u}_2 \quad (48)$$

만약 u_1, u_2 가 단위벡터이면 다음과 같다.

$$\hat{y} = \text{proj}_L y = (\mathbf{y} \cdot \mathbf{u}_1) \mathbf{u}_1 + (\mathbf{y} \cdot \mathbf{u}_2) \mathbf{u}_2 \quad (49)$$

프로젝션은 각각 직교기저벡터에 독립적으로 적용된다.

39 Orthogonal Projection when $y \in W$

만약 2차원 부분공간 $W = \text{Span}\{u_1, u_2\}$ 에 y 가 포함되어 있다고 하면 프로젝션된 벡터 \hat{y} 는 다음과 같이 구할 수 있다.

$$\hat{y} = \text{proj}_L y = y = \frac{\mathbf{y} \cdot \mathbf{u}_1}{\mathbf{u}_1 \cdot \mathbf{u}_1} \mathbf{u}_1 + \frac{\mathbf{y} \cdot \mathbf{u}_2}{\mathbf{u}_2 \cdot \mathbf{u}_2} \mathbf{u}_2 \quad (50)$$

만약 u_1, u_2 가 단위벡터이면 다음과 같다.

$$\hat{\mathbf{y}} = \text{proj}_L \mathbf{y} = \mathbf{y} = (\mathbf{y} \cdot \mathbf{u}_1)\mathbf{u}_1 + (\mathbf{y} \cdot \mathbf{u}_2)\mathbf{u}_2 \quad (51)$$

해는 \mathbf{y} 가 부분공간 W 에 포함되어 있지 않은 경우와 동일하다.

40 Transformation: Orthogonal Projection

부분공간 W 의 정규직교기저벡터 $\mathbf{u}_1, \mathbf{u}_2$ 가 있고 \mathbf{b} 를 부분공간 W 에 프로젝션시킨 점 $\hat{\mathbf{b}}$ 의 변환을 생각해보면

$$\begin{aligned}\hat{\mathbf{b}} &= f(\mathbf{b}) = (\mathbf{b} \cdot \mathbf{u}_1)\mathbf{u}_1 + (\mathbf{b} \cdot \mathbf{u}_2)\mathbf{u}_2 \\ &= (\mathbf{u}_1^\top \mathbf{b})\mathbf{u}_1 + (\mathbf{u}_2^\top \mathbf{b})\mathbf{u}_2 \\ &= \mathbf{u}_1(\mathbf{u}_1^\top \mathbf{b}) + \mathbf{u}_2(\mathbf{u}_2^\top \mathbf{b}) \\ &= (\mathbf{u}_1 \mathbf{u}_1^\top) \mathbf{b} + (\mathbf{u}_2 \mathbf{u}_2^\top) \mathbf{b} \\ &= (\mathbf{u}_1 \mathbf{u}_1^\top + \mathbf{u}_2 \mathbf{u}_2^\top) \mathbf{b} \\ &= [\mathbf{u}_1 \ \mathbf{u}_2] \begin{bmatrix} \mathbf{u}_1^\top \\ \mathbf{u}_2^\top \end{bmatrix} \mathbf{b} = \mathbf{U} \mathbf{U}^\top \mathbf{b} \Rightarrow \text{Linear Transformation!}\end{aligned} \quad (52)$$

41 Orthogonal Projection Perspective

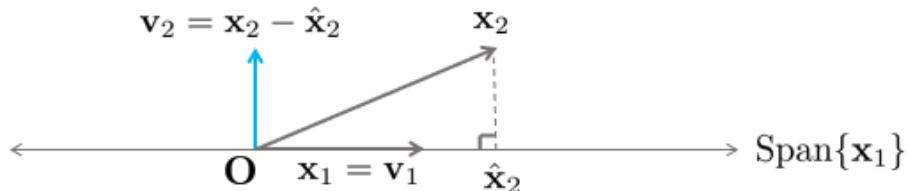
정규직교인 열벡터를 가지는 행렬 $\mathbf{A} = \mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2]$ 가 있을 때 \mathbf{b} 벡터를 $\text{Col } \mathbf{A}$ 공간으로 정사영시키는 경우

$$\hat{\mathbf{b}} = \mathbf{A} \hat{\mathbf{x}} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b} = f(\mathbf{b}) \quad (53)$$

행렬 $\mathbf{C} = \mathbf{A}^\top \mathbf{A}$ 는 $\mathbf{C} = \begin{bmatrix} \mathbf{u}_1^\top \\ \mathbf{u}_2^\top \end{bmatrix} [\mathbf{u}_1 \ \mathbf{u}_2] = \mathbf{I}$ 와 같은 성질을 지니게 되고 따라서 다음과 같은 공식이 성립한다.

$$\hat{\mathbf{b}} = \mathbf{A} \hat{\mathbf{x}} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b} = \mathbf{A}(\mathbf{I})^{-1} \mathbf{A}^\top \mathbf{b} = \mathbf{A} \mathbf{A}^\top \mathbf{b} = \mathbf{U} \mathbf{U}^\top \mathbf{b} \quad (54)$$

42 Gram-Schmidt Orthogonalization



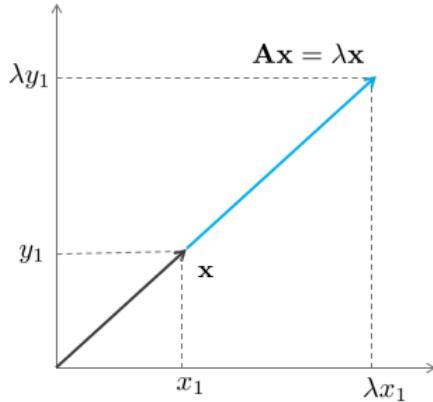
벡터 $\mathbf{x}_1 = \begin{bmatrix} 3 \\ 6 \\ 0 \end{bmatrix}, \mathbf{x}_2 = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$ 로 인해 $\text{Span}\{\mathbf{x}_1\}$ 되는 부분공간 $Wx_1 = \text{Span}[\mathbf{x}_1 \ \mathbf{x}_2]$ 가 있을 때 두 벡터의 내적 $\mathbf{x}_1 \cdot \mathbf{x}_2 = 15 \neq 0$ 으로 두 벡터는 수직이 아니다.

이 때 벡터 $\mathbf{v}_1 = \mathbf{x}_1$ 이라고 하고 \mathbf{v}_2 를 \mathbf{x}_1 에 수직인 \mathbf{x}_2 의 성분이라고 했을 때

$$\mathbf{v}_2 = \mathbf{x}_2 - \frac{\mathbf{x}_2 \cdot \mathbf{x}_1}{\mathbf{x}_1 \cdot \mathbf{x}_1} \mathbf{x}_1 = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} - \frac{15}{45} \begin{bmatrix} 3 \\ 6 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} \quad (55)$$

가 된다. 이 때 벡터 $\mathbf{v}_1, \mathbf{v}_2$ 는 부분공간 W 의 직교기저벡터가 된다.

43 Eigenvectors and Eigenvalues



정방행렬 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 에 대한 고유벡터(eigenvector)는 $\mathbf{Ax} = \lambda\mathbf{x}$ 를 만족하는 0이 아닌 벡터 $\mathbf{x} \in \mathbb{R}^n$ 을 말한다. 이 때 λ 는 행렬 \mathbf{A} 의 고유값(eigenvalue)이라고 한다.

$\mathbf{Ax} = \lambda\mathbf{x}$ 는 다음과 같이 다시 나타낼 수 있다.

$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = 0 \quad (56)$$

이 때, 위 시스템이 \mathbf{x} 가 0이 아닌 비자명해를 가지고 있는 경우에만 λ 값이 행렬 \mathbf{A} 에 대한 고유값이 된다. 위와 같은 동차 선형시스템이 비자명해를 가지기 위해서는 $\mathbf{A} - \lambda\mathbf{I}$ 가 선형의존(Linearily Dependent) 해야 무수히 많은 해를 가진다.

44 Null Space

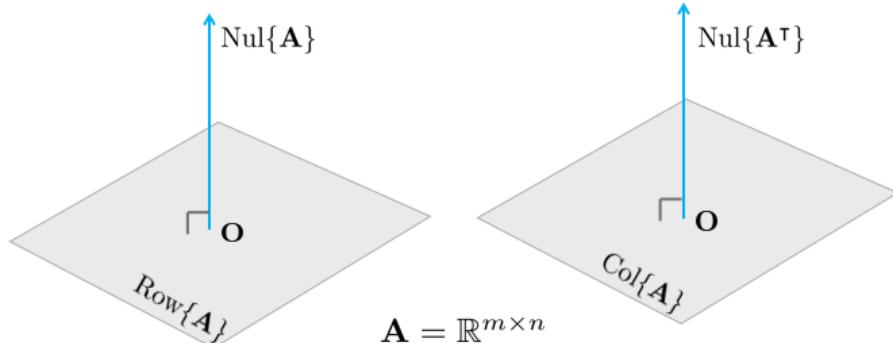
행렬 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 의 동차 선형시스템(Homogeneous Linear System) $\mathbf{Ax} = 0$ 의 해 집합을 영공간(Null Space)라고 한다. $\text{Nul } \mathbf{A}$ 로 표기한다.

$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \vdots \\ \mathbf{a}_m^\top \end{bmatrix}$ 일 때 벡터 \mathbf{x} 는 다음을 만족해야 한다.

$$\mathbf{a}_1^\top \mathbf{x} = \mathbf{a}_2^\top \mathbf{x} = \cdots = \mathbf{a}_m^\top \mathbf{x} = 0 \quad (57)$$

즉, \mathbf{x} 는 모든 \mathbf{A} 의 행벡터(Row Vector)과 직교해야 한다.

45 Orthogonal Complement



벡터 \mathbf{z} 가 부분공간 $W \in \mathbb{R}^n$ 의 모든 벡터와 직교하면 \mathbf{z} 는 부분공간 W 와 직교한다고 말할 수 있다. 부분공간 W 와 직교하는 모든 벡터 \mathbf{z} 의 집합을 직교여공간(Orthogonal Complement)라고 부르며 W^\perp 로 표시한다.

부분공간 W 의 직교여공간 W^\perp 에 위치한 벡터 $\mathbf{x} \in \mathbb{R}^n$ 는 부분공간 W 를 Span하는 모든 벡터들과 직교한다.

W^\perp is a subspace of \mathbb{R}^n .

$$\begin{aligned}\text{Nul}\mathbf{A} &= (\text{Row}\mathbf{A})^\perp \\ \text{Nul}\mathbf{A}^\top &= (\text{Col}\mathbf{A})^\perp\end{aligned}\tag{58}$$

46 Characteristic Equation

방정식 $(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = 0$ 이 비자명해를 갖기 위해서는 $(\mathbf{A} - \lambda\mathbf{I})$ 행렬이 선형의존이어야 하고 이는 곧 역행렬이 존재하지 않아야 하는 것과 동치(Equivalent)이다. 만약 $(\mathbf{A} - \lambda\mathbf{I})$ 이 역행렬이 존재한다면 \mathbf{x} 는 자명해 이외에는 갖지 못한다.

$$\begin{aligned}(\mathbf{A} - \lambda\mathbf{I})^{-1}(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} &= (\mathbf{A} - \lambda\mathbf{I})^{-1}0 \\ \mathbf{x} &= 0\end{aligned}\tag{59}$$

따라서 행렬 \mathbf{A} 에 대하여 고유값과 고유벡터가 존재하기 위해서는 다음의 방정식이 항상 성립해야 한다.

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0\tag{60}$$

위 방정식을 행렬 \mathbf{A} 의 특성방정식(Characteristic Equation)이라고 부른다.

47 Eigenspace

$(\mathbf{A} - \lambda\mathbf{x})\mathbf{x} = 0$ 에서 $(\mathbf{A} - \lambda\mathbf{x})$ 의 영공간(Null Space)을 고유값 λ 에 대한 고유공간(Eigenspace)라고 한다. λ 에 대한 고유공간의 차원이 1 이상인 경우, 고유공간 내에 있는 모든 벡터들에 대하여 다음이 성립한다.

$$T(\mathbf{x}) = \mathbf{Ax} = \lambda\mathbf{x}\tag{61}$$

48 Diagonalization

정방행렬 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 이 주어졌고 $\mathbf{V} \in \mathbb{R}^{n \times n}$ 이고 $\mathbf{D} \in \mathbb{R}^{n \times n}$ 일 때

$$\mathbf{D} = \mathbf{V}^{-1}\mathbf{AV}\tag{62}$$

위와 같은 공식이 성립한다면 이를 정방행렬 \mathbf{A} 의 대각화(Diagonalization)라고 한다. 대각화는 모든 경우에 대해서 항상 가능한 것은 아니다. 행렬 \mathbf{A} 가 대각화되기 위해서는 역행렬이 존재하는 행렬 \mathbf{V} 가 존재해야 한다. 행렬 \mathbf{V} 가 역행렬이 존재하기 위해서는 \mathbf{V} 는 행렬 \mathbf{A} 와 같은 $\mathbb{R}^{n \times n}$ 크기의 정방행렬이어야 하고 n 개의 선형독립인 열벡터를 가지고 있어야 한다. 이 때, \mathbf{V} 의 각 열은 행렬 \mathbf{A} 의 고유벡터가 된다. 만약 행렬 \mathbf{V} 가 존재하는 경우 행렬 \mathbf{A} 는 대각화 가능(Diagonalizable)하다고 한다.

49 Finding \mathbf{V} and \mathbf{D}

대각화 공식은 다음과 같이 다시 작성할 수 있다.

$$\mathbf{D} = \mathbf{V}^{-1}\mathbf{AV} \Rightarrow \mathbf{VD} = \mathbf{AV}\tag{63}$$

$$\text{이 때, } \mathbf{V} = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_n] \text{이고 } \mathbf{D} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix} \text{이라고 하면}$$

$$\begin{aligned}
 \mathbf{AV} &= \mathbf{A} [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_n] = [\mathbf{Av}_1 \quad \mathbf{Av}_2 \quad \cdots \quad \mathbf{Av}_n] \\
 \mathbf{VD} &= [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_n] \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix} \\
 &= [\lambda_1 \mathbf{v}_1 \quad \lambda_2 \mathbf{v}_2 \quad \cdots \quad \lambda_n \mathbf{v}_n] \\
 \mathbf{AV} = \mathbf{VD} &\Leftrightarrow [\mathbf{Av}_1 \quad \mathbf{Av}_2 \quad \cdots \quad \mathbf{Av}_n] = [\lambda_1 \mathbf{v}_1 \quad \lambda_2 \mathbf{v}_2 \quad \cdots \quad \lambda_n \mathbf{v}_n]
 \end{aligned} \tag{64}$$

위 공식과 같아

$$\mathbf{Av}_1 = \lambda_1 \mathbf{v}_1, \mathbf{Av}_2 = \lambda_2 \mathbf{v}_2, \dots, \mathbf{Av}_n = \lambda_n \mathbf{v}_n \tag{65}$$

각각의 열이 모두 동일해야 한다. 즉, 벡터 \mathbf{v}_i 는 행렬 \mathbf{A} 에 대한 고유벡터가 되어야 하고 스칼라 λ_i 는 행렬 \mathbf{A} 에 대한 고유값이 되어야 한다. 이에 따라 대각행렬 \mathbf{D} 는 고유값들을 대각성분으로 포함하고 있는 행렬이 된다. 결론적으로 정방행렬 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 가 대각화 가능한가 안한가에 대한 질문은 n 개의 고유벡터가 존재하는가 안하는가에 대한 질문과 동치이다.

50 Eigendecomposition

정방행렬 \mathbf{A} 가 대각화 가능한 경우 $\mathbf{D} = \mathbf{V}^{-1} \mathbf{AV}$ 공식이 성립한다. 이 공식을 다시 작성하면 다음과 같다.

$$\mathbf{A} = \mathbf{VDV}^{-1} \tag{66}$$

이를 행렬 \mathbf{A} 에 대한 고유값 분해(Eigendecomposition)라고 한다. 행렬 \mathbf{A} 가 대각화 가능하다는 의미는 행렬 \mathbf{A} 가 고유값 분해 가능하다는 말과 동치이다.

51 Linear Transformation via Eigendecomposition

정방행렬 \mathbf{A} 가 대각화 가능한 경우 $\mathbf{A} = \mathbf{VDV}^{-1}$ 과 같이 고유값 분해가 가능하다. 이 때 선형 변환 $T(\mathbf{x}) = \mathbf{Ax}$ 을 생각해보면 다음과 같이 표현할 수 있다.

$$T(\mathbf{x}) = \mathbf{Ax} = \mathbf{VDV}^{-1}\mathbf{x} = \mathbf{V}(\mathbf{D}(\mathbf{V}^{-1}\mathbf{x})) \tag{67}$$

51.1 Change of Basis

예를 들어 $\mathbf{Av}_1 = -1\mathbf{v}_1, \mathbf{Av}_2 = 2\mathbf{v}_2$ 가 성립한다고 가정하고 $T(\mathbf{x}) = \mathbf{Ax} = \mathbf{VDV}^{-1}\mathbf{x} = \mathbf{V}(\mathbf{D}(\mathbf{V}^{-1}\mathbf{x}))$ 에서 $\mathbf{y} = \mathbf{V}^{-1}\mathbf{x}$ 라고 가정하면

$$\mathbf{V}\mathbf{y} = \mathbf{x} \tag{68}$$

의 관계가 성립한다. 이 때, 벡터 \mathbf{y} 는 벡터 \mathbf{x} 의 고유벡터 $\{\mathbf{v}_1, \mathbf{v}_2\}$ 에 대한 새로운 좌표를 의미한다.

$$\mathbf{x} = \begin{bmatrix} 4 \\ 3 \end{bmatrix} = 4 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 3 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 3 \end{bmatrix} = \mathbf{V}\mathbf{y} = [\mathbf{v}_1 \quad \mathbf{v}_2] \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = 2\mathbf{v}_1 + 1\mathbf{v}_2 \Rightarrow \mathbf{y} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \tag{69}$$

51.2 Element-wise Scaling

위 과정을 통해 \mathbf{y} 값을 구하고 나면 $T(\mathbf{x}) = \mathbf{V}(\mathbf{D}(\mathbf{V}^{-1}\mathbf{x}))$ 는 $T(\mathbf{x}) = \mathbf{V}(\mathbf{D}\mathbf{y})$ 로 표현할 수 있다. 이 때 $\mathbf{z} = \mathbf{D}\mathbf{y}$ 라고 하면 벡터 \mathbf{z} 는 단순히 벡터 \mathbf{y} 를 행렬의 대각 원소의 크기만큼 스케일링한 벡터가 된다.

51.3 Back to Original Basis

위 과정까지 진행했으면 $T(\mathbf{x}) = \mathbf{V}(\mathbf{D}\mathbf{y}) = \mathbf{V}\mathbf{z}$ 와 같이 나타낼 수 있고 이 때 벡터 \mathbf{z} 는 여전히 새로운 기저 벡터 $\{\mathbf{v}'_1, \mathbf{v}'_2\}$ 를 기반으로 하면 좌표가 된다. **$\mathbf{V}\mathbf{z}$ 연산은 벡터 \mathbf{z} 를 다시 원래 기저벡터의 좌표로 변환하는 역할을 한다.** 벡터 $\mathbf{V}\mathbf{z}$ 는 기존의 기저벡터 $\{\mathbf{v}_1, \mathbf{v}_2\}$ 의 선형결합이 된다.

$$\mathbf{V}\mathbf{z} = [\mathbf{v}_1 \quad \mathbf{v}_2] \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \mathbf{v}_1 z_1 + \mathbf{v}_2 z_2 \quad (70)$$

지금까지의 과정을 **고유값 분해를 통한 선형 변환**이라고 한다.

52 Linear Transformation via \mathbf{A}^k

여러번의 변환이 중첩된 $\mathbf{A} \times \mathbf{A} \times \cdots \times \mathbf{A}\mathbf{x} = \mathbf{A}^k\mathbf{x}$ 를 생각해보자. 이 때, 행렬 \mathbf{A} 가 대각화 가능하다면 \mathbf{A} 를 고유값 분해할 수 있고 이 때, \mathbf{A}^k 는 다음과 같이 분해할 수 있다.

$$\mathbf{A}^k = (\mathbf{V}\mathbf{D}\mathbf{V}^{-1})(\mathbf{V}\mathbf{D}\mathbf{V}^{-1}) \cdots (\mathbf{V}\mathbf{D}\mathbf{V}^{-1}) = \mathbf{V}\mathbf{D}^k\mathbf{V}^{-1} \quad (71)$$

이 때 \mathbf{D}^k 는 다음과 같이 표현된다.

$$\mathbf{D}^k = \begin{bmatrix} \lambda_1^k & 0 & \cdots & 0 \\ 0 & \lambda_2^k & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n^k \end{bmatrix} \quad (72)$$

53 Geometric Multiplicity and Algebraic Multiplicity

정방행렬 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 이 있을 때 \mathbf{A} 가 대각화 가능한지 안한지 판단을 해야하는 경우 일반적으로 판별식을 사용하여 판단한다.

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0 \quad (73)$$

예를 들어 $n = 5$ 인 정방행렬 \mathbf{A} 가 있을 때, $\det(\mathbf{A} - \lambda\mathbf{I})$ 는 5차 다항식이 나오게 된다. 5차 다항식은 일반적으로 5개의 해를 가지고 있지만 실수만 고려하는 경우 5개의 해가 계산되지 않을 수 있다. 즉, **실근이 5개가 나오지 않는 경우 $n = 5$ 개의 선형독립인 고유벡터가 나오지 않으므로 대각화가 불가능하다.**

만약 실근 중 중근이 포함되는 경우, 예를 들어 $(\lambda - 2)^2(\lambda - 3) = 0$ 과 같이 $\lambda = 2$ 가 중근인 경우, $\lambda = 2$ 로 인해 생성되는 고유공간(Eigenspace)의 차원이 최대 $\lambda = 2$ 가 가지는 중근의 개수까지 가질 수 있다. 중근이 아닌 일반 실근의 경우 최대 1차원의 고유공간을 가질 수 있다. 즉, 중근이 포함된 경우 고유공간의 차원이 최대 $n = 5$ 까지 생성될 수 있는데 $n = 5$ 를 만족하지 못하는 경우에는 대각화가 불가능하다.

이와 같이 대수적으로 판별식을 인수분해했을 때, 중근이 생기는 경우 중근의 대수 중복도(Algebraic Multiplicity)와 이로 인해 Span되는 고유공간의 기하 중복도(Geometric Multiplicity)가 일치해야 n 개의 독립적인 고유벡터가 생성될 수 있고 행렬 \mathbf{A} 의 대각화가 가능하다.

54 Singular Value Decomposition

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\top =$$

$(\mathbf{A} \in \mathbb{R}^{m \times n})$

행렬 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 이 주어졌을 때 특이값 분해(Singular Value Decomposition, SVD)는 다음과 같이 나타낼 수 있다.

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top \quad (74)$$

이 때, $\mathbf{U} \in \mathbb{R}^{m \times m}$, $\mathbf{V} \in \mathbb{R}^{n \times n}$ 인 행렬이며 이들은 각 열이 Col \mathbf{A} 와 Row \mathbf{A} 에 대한 정규직교기저벡터(Orthonormal Basis)로 구성되어 있다. $\Sigma \in \mathbb{R}^{m \times n}$ 은 대각행렬이며 대각 성분들이 $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min(m,n)}$ 특이값이며 큰 값부터 내림차순으로 정렬된 행렬이다.

55 SVD as Sum of Outer Products

행렬 \mathbf{A} 는 다음과 같이 Outer Products의 합으로 표현할 수 있다.

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top = \sum_{i=1}^n \sigma_i \mathbf{u}_i \mathbf{v}_i^\top, \quad \text{where } \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \quad (75)$$

이 때 위 식을 다시 행렬로 합성하면 $\mathbf{U} \in \mathbb{R}^{m \times m} \rightarrow \mathbf{U}' \in \mathbb{R}^{m \times n}$ 그리고 $\mathbf{D} \in \mathbb{R}^{m \times n} \rightarrow \mathbf{D}' \in \mathbb{R}^{n \times n}$ 과 같이 행렬 \mathbf{V}^\top 의 차원에 맞게 다시 합성할 수 있는데 이를 **Reduced Form of SVD**이라고 한다.

$$\mathbf{A} = \mathbf{U}'\mathbf{D}'\mathbf{V}^\top \quad (76)$$

56 Another Perspective of SVD

행렬 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 에 대해 Gram-Schmidt Orthogonalization을 사용하면 Col \mathbf{A} 에 대한 정규직교기저벡터 $\mathbf{u}_1, \dots, \mathbf{u}_n$ 과 Row \mathbf{A} 에 대한 정규직교기저벡터 $\mathbf{v}_1, \dots, \mathbf{v}_n$ 을 구할 수 있다. 하지만 이렇게 계산한 정규직교기저벡터 $\mathbf{u}_i, \mathbf{v}_i$ 는 유일하지 않다.

Reduced Form of SVD를 사용하면 행렬 $\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_n] \in \mathbb{R}^{m \times n}$ 과 $\mathbf{V} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_n] \in \mathbb{R}^{n \times n}$ 그리고 $\Sigma = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \sigma_n \end{bmatrix} \in \mathbb{R}^{n \times n}$ 일 때

$$\begin{aligned} \mathbf{AV} &= \mathbf{A}[\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_n] = [\mathbf{Av}_1 \ \mathbf{Av}_2 \ \dots \ \mathbf{Av}_n] \\ &\quad \left[\begin{array}{cccc} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \sigma_n \end{array} \right] \\ \mathbf{U}\Sigma &= [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_n] \left[\begin{array}{cccc} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \sigma_n \end{array} \right] \\ &\quad [\sigma_1 \mathbf{u}_1 \ \sigma_2 \mathbf{u}_2 \ \dots \ \sigma_n \mathbf{u}_n] \\ \mathbf{AV} &= \mathbf{U}\Sigma \Leftrightarrow [\mathbf{Av}_1 \ \mathbf{Av}_2 \ \dots \ \mathbf{Av}_n] = [\sigma_1 \mathbf{u}_1 \ \sigma_2 \mathbf{u}_2 \ \dots \ \sigma_n \mathbf{u}_n] \end{aligned} \quad (77)$$

위 식을 간결하게 나타내면 다음과 같다.

$$\mathbf{AV} = \mathbf{U}\Sigma \Leftrightarrow \mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top \quad (78)$$

57 Computing SVD

행렬 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 에 대하여 \mathbf{AA}^\top 와 $\mathbf{A}^\top\mathbf{A}$ 는 다음과 같이 고유값분해할 수 있다.

$$\begin{aligned} \mathbf{AA}^\top &= \mathbf{U}\Sigma\mathbf{V}^\top\mathbf{V}\Sigma^\top\mathbf{U}^\top = \mathbf{U}\Sigma\Sigma^\top\mathbf{U}^\top = \mathbf{U}\Sigma^2\mathbf{U}^\top \\ \mathbf{A}^\top\mathbf{A} &= \mathbf{V}\Sigma^\top\mathbf{U}^\top\mathbf{U}\Sigma\mathbf{V}^\top = \mathbf{V}\Sigma^\top\Sigma\mathbf{U}^\top = \mathbf{V}\Sigma^2\mathbf{V}^\top \end{aligned} \quad (79)$$

이 때 계산되는 행렬 \mathbf{U}, \mathbf{V} 은 직교하는 고유벡터를 각 열의 성분으로 하는 행렬이며 대각행렬 Σ^2 의 각 성분은 항상 0보다 큰 양수의 값을 가진다. 그리고 \mathbf{AA}^\top 와 $\mathbf{A}^\top\mathbf{A}$ 를 통해 계산되는 Σ^2 의 값은 동일하다.

58 Diagonalization of Symmetric Matrices

일반적으로 정방행렬 $A \in \mathbb{R}^{n \times n}$ 이 n 개의 선형독립인 고유벡터를 가지고 있을 경우 대각화 가능하다. 그리고 대칭행렬 $S \in \mathbb{R}^{n \times n}$, $S^T = S$ 는 항상 대각화 가능하다. 추가적으로 대칭행렬 S 의 고유벡터는 항상 서로에게 직교하므로 직교대각화(Orthogonally Diagonalizable)가 가능하다.

59 Spectral Theorem of Symmetric Matrices

$S^T = S$ 를 만족하는 대칭행렬 S 가 주어졌을 때 S 는 n 개의 중근을 포함한 실수의 고유값이 존재한다. 또한, 고유공간의 차원은 기하 중복도(Algebraic Multiplicity)와 기하 중복도(Geometric Multiplicity)와 같아야 한다. 서로 다른 λ 값 들에 대한 고유공간들은 서로 직교한다. 결론적으로 대칭행렬 S 은 직교대각화가 가능하다.

60 Spectral Decomposition

대칭행렬 S 의 고유값 분해는 **Spectral Decomposition**이라고 불린다. 이는 다음과 같이 나타낼 수 있다.

$$\begin{aligned} S &= UDU^{-1} = UDU^T = [u_1 \ u_2 \ \cdots \ u_n] \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix} \begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_n^T \end{bmatrix} \\ &= [\lambda_1 u_1 \ \lambda_2 u_2 \ \cdots \ \lambda_n u_n] \begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_n^T \end{bmatrix} \\ &= \lambda_1 u_1 u_1^T + \lambda_2 u_2 u_2^T + \cdots + \lambda_n u_n u_n^T \end{aligned} \tag{80}$$

위 식에서 각 항 $\lambda_i u_j u_j^T$ 은 u_j 에 의해 Span된 부분공간에 프로젝션된 다음 고유값 λ_i 만큼 스케일된 벡터로 볼 수 있다.

61 Positive (Semi-)Definite Matrices

정방행렬 $A \in \mathbb{R}^{n \times n}$ 이 있을 때 0이 아닌 모든 벡터 $\forall x \neq 0$ 에 대하여 $x^T A x > 0$ 을 만족하는 경우 A 를 **Positive Definite** 행렬이라고 한다. 만약 $x^T A x \geq 0$ 인 경우 **Positive Semi-Definite** 행렬이라고 한다.

정방행렬 A 가 Postivie Definite인 경우 **A의 고유값은 항상 모두 양수이다.**

62 Symmetric Positive Definite Matrices

행렬 $S \in \mathbb{R}^{n \times n}$ 이 대칭이면서 Positive Definite인 경우 Spectral Decomposition의 모든 고유값은 항상 양수가 된다.

$$\begin{aligned} S &= UDU^{-1} = UDU^T = [u_1 \ u_2 \ \cdots \ u_n] \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix} \begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_n^T \end{bmatrix} \\ &= \lambda_1 u_1 u_1^T + \lambda_2 u_2 u_2^T + \cdots + \lambda_n u_n u_n^T \\ &\text{where, } \lambda_j > 0, \forall j = 1, \dots, n \end{aligned} \tag{81}$$

63 Back to Computing SVD

행렬 A 에 대하여 $AA^T = A^T A = S$ 인 대칭행렬이 존재할 때 S 가 Positive (Semi-)Definite한 경우

$$\begin{aligned} \mathbf{x}^T \mathbf{A} \mathbf{A}^T \mathbf{x} &= (\mathbf{A}^T \mathbf{x})^T (\mathbf{A}^T \mathbf{x}) = \|\mathbf{A}^T \mathbf{x}\| \geq 0 \\ \mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x} &= (\mathbf{A} \mathbf{x})^T (\mathbf{A} \mathbf{x}) = \|\mathbf{A} \mathbf{x}\|^2 \geq 0 \end{aligned} \quad (82)$$

즉, $\mathbf{A} \mathbf{A}^T = \mathbf{U} \Sigma^2 \mathbf{U}^T$ 와 $\mathbf{A}^T \mathbf{A} = \mathbf{V} \Sigma^2 \mathbf{V}^T$ 에서 Σ^2 의 값은 항상 양수가 된다.

임의의 각 행렬 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 에 대하여 특이값 분해는 언제나 존재한다. $\mathbf{A} \in \mathbb{R}^{n \times n}$ 행렬의 경우 고유값 분해가 존재하지 않을 수 있지만 특이값 분해는 항상 존재한다. 대칭이면서 동시에 Positive Definite인 정방 행렬 $\mathbf{S} \in \mathbb{R}^{n \times n}$ 은 항상 고유값 분해값이 존재하며 이는 특이값 분해와 동일하다.

64 Eigendecomposition in Machine Learning

일반적으로 머신러닝에서는 대칭이고 Positive Definite인 행렬을 다룬다. 예를 들면, $\mathbf{A} \in \mathbb{R}^{10 \times 3}$ 인 행렬이 있고 각 열은 사람을 의미하고 각 행은 Feature를 의미한다고 가정했을 때, $\mathbf{A}^T \mathbf{A} \in \mathbb{R}^{3 \times 3}$ 는 각 사람들 간 유사도를 의미하고 $\mathbf{A} \mathbf{A}^T \in \mathbb{R}^{10 \times 10}$ 는 각 Feature들의 상관관계를 의미한다. 이 때, $\mathbf{A} \mathbf{A}^T$ 는 주성분분석 (Principal Component Analysis)에서 Covariance Matrix를 구할 때 사용된다.

65 Low Rank Approximation of a Matrix

행렬 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 이 주어졌을 때 예를 들어, \mathbf{A} 의 원래 rank가 r 일 때, 행렬 \mathbf{A} 에서 rank를 r 이하를 가진 근사행렬 $\hat{\mathbf{A}}$ 을 찾는 Low Rank Approximation을 수행할 수 있다.

$$\begin{aligned} \hat{\mathbf{A}}_r &= \arg \min_{\hat{\mathbf{A}}_r} \|\mathbf{A} - \mathbf{A}_r\|_F, \quad \text{subject to } \text{rank} \mathbf{A}_r \leq r \\ \hat{\mathbf{A}}_r &= \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^T \quad \text{where, } \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \end{aligned} \quad (83)$$

66 Dimension Reducing Transformation

Feature-by-data item 행렬 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 이 주어졌을 때 $\mathbf{G} \in \mathbb{R}^{m \times r}, r < m$ 인 변환 $\mathbf{G}^T : \mathbf{x} \in \mathbb{R}^m \mapsto \mathbf{y} \in \mathbb{R}^r$ 을 생각해보면

$$\mathbf{y}_i = \mathbf{G}^T \mathbf{a}_i \quad (84)$$

가 성립하고 \mathbf{G} 의 각 열들은 정규직교벡터이며 데이터의 유사도 행렬 $\mathbf{S} = \mathbf{A}^T \mathbf{A}$ 의 유사도를 보존하는 변환 \mathbf{G} 를 차원 축소 변환(Dimension-Reducing Transformation)이라고 한다.

$$\begin{aligned} \mathbf{Y} &= \mathbf{G}^T \mathbf{A} \\ \mathbf{Y}^T \mathbf{Y} &= (\mathbf{G}^T \mathbf{A})^T \mathbf{G}^T \mathbf{A} = \mathbf{A}^T \mathbf{G} \mathbf{G}^T \mathbf{A} \end{aligned} \quad (85)$$

이 때 차원축소변환 $\hat{\mathbf{G}}$ 은 다음과 같이 추정할 수 있다.

$$\hat{\mathbf{G}} = \arg \min_{\mathbf{G}} \|S - \mathbf{A}^T \mathbf{G} \mathbf{G}^T \mathbf{A}\|_F \quad \text{subject to } \mathbf{G}^T \mathbf{G} = \mathbf{I}_k \quad (86)$$

주어진 행렬 $\mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^T = \sum_{i=1}^n \sigma_i \mathbf{u}_i \mathbf{v}_i^T$ 에 대하여 최적의 해는 다음과 같다.

$$\hat{\mathbf{G}} = \mathbf{U}_r = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \dots \quad \mathbf{u}_r] \quad (87)$$

67 Derivative of multi-variable function

67.1 Gradient

임의의 벡터 $\mathbf{x} \in \mathbb{R}^n$ 에 대하여 $f(\mathbf{x}) \in \mathbb{R}$ 를 만족하는 다변수 스칼라 함수 $f(\mathbf{x})$ 가 주어졌다고 하자.

$$f : \mathbb{R}^n \mapsto \mathbb{R} \quad (88)$$

$f(\mathbf{x})$ 에 대한 1차 편미분은 벡터가 되고 이는 그레디언트(gradient)라고 불린다.

$$\boxed{\nabla f = \left(\frac{\partial f}{\partial x_1} \quad \dots \quad \frac{\partial f}{\partial x_n} \right) \in \mathbb{R}^{1 \times n}} \quad (89)$$

67.2 Jacobian matrix

임의의 벡터 $\mathbf{x} \in \mathbb{R}^n$ 에 대하여 $f(\mathbf{x}) \in \mathbb{R}^m$ 를 만족하는 다변수 벡터 함수 $f(\mathbf{x})$ 가 주어졌다고 하자.

$$f : \mathbb{R}^n \mapsto \mathbb{R}^m \quad (90)$$

이 때, $f(\cdot)$ 의 1차 편미분은 행렬이 되고 이를 특별히 자코비안(jacobian) 행렬이라고 한다.

$$\boxed{\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{m \times n}} \quad (91)$$

이를 통해 자코비안 행렬의 각 행벡터(row vector)는 함수 $f_m(\cdot)$ 에 대한 그레디언트라는 것을 알 수 있다. 위 식을 미소변화량 \mathbf{h} 를 사용하여 나타내면 다음과 같다.

$$\mathbf{J} = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \triangleq \lim_{\mathbf{h} \rightarrow 0} \frac{f(\mathbf{x} + \mathbf{h}) - f(\mathbf{x})}{\mathbf{h}} \in \mathbb{R}^{m \times n} \quad (92)$$

SLAM에서 자코비안은 에러 $\mathbf{e}(\mathbf{x})$ 를 최적화할 때 사용된다. SLAM에서 최적화하고자 하는 에러는 일반적으로 비선형 함수로 구성되어 있으며 크기가 작기 때문에 에러의 변화량 $\mathbf{e}(\mathbf{x} + \Delta \mathbf{x})$ 를 그대로 사용하지 않고 테일러 전개하여 근사식 $\mathbf{e}(\mathbf{x}) + \mathbf{J} \Delta \mathbf{x}$ 으로 표현하게 되는데 이 때 에러에 대한 자코비안 \mathbf{J} 가 유도된다. 그리고 근사식을 바탕으로 유도한 에러의 최적 증분량 $\Delta \mathbf{x}^* = (\mathbf{J}^\top \mathbf{J})^{-1} \mathbf{J}^\top \mathbf{b}$ 이 자코비안을 통해 구해지기 때문에 SLAM에서는 자코비안이 필수적으로 사용된다. 자세한 내용은 [SLAM] Errors and Jacobian Derivations for SLAM 정리 포스트를 참조하면 된다.

67.2.1 Toy example

예를 들어 다음과 같은 3개의 연립 방정식이 주어졌다고 하자.

$$f(\mathbf{x}) = \begin{cases} f_1(\mathbf{x}) = ax^2 + 2bx + cy \\ f_2(\mathbf{x}) = dx^3 + ex \\ f_3(\mathbf{x}) = fx + gy^2 + hy \end{cases} \quad (93)$$

$\mathbf{x} = (x, y)$ 를 의미한다. 위 함수는 다음과 같이 쓸 수 있다.

$$f : \mathbb{R}^2 \mapsto \mathbb{R}^3 \quad (94)$$

자코비안의 정의에 따라 이를 아래와 같이 쓸 수 있다.

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix} = \begin{bmatrix} 2ax + 2b & c \\ 3dx^2 + e & 0 \\ f & 2gy + h \end{bmatrix} \in \mathbb{R}^{3 \times 2} \quad (95)$$

67.3 Hessian matrix

임의의 벡터 $\mathbf{x} \in \mathbb{R}^n$ 에 대하여 $f(\mathbf{x}) \in \mathbb{R}$ 를 만족하는 다변수 스칼라 함수 $f(\mathbf{x})$ 가 주어졌다고 하자.

$$f : \mathbb{R}^n \mapsto \mathbb{R} \quad (96)$$

이 때, $f(\cdot)$ 의 2차 편미분은 행렬이 되고 이를 특별히 헤시안(hessian) 행렬이라고 한다. 헤시안 행렬은 일반적으로 대칭행렬의 형태를 띠고 있으며 다변수 벡터 함수가 아닌 다변수 스칼라 함수에 대한 2차 미분임에 유의한다.

$$\boxed{\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \in \mathbb{R}^{n \times n}} \quad (97)$$

67.4 Laplacian

임의의 벡터 $\mathbf{x} \in \mathbb{R}^n$ 에 대하여 $f(\mathbf{x}) \in \mathbb{R}$ 를 만족하는 다변수 스칼라 함수 $f(\mathbf{x})$ 가 주어졌다고 하자.

$$f : \mathbb{R}^n \mapsto \mathbb{R} \quad (98)$$

$f(\mathbf{x})$ 에 대한 라플라시안(laplacian)은 각 입력 벡터에 따른 2차 편미분의 합으로 정의된다.

$$\nabla^2 \mathbf{f} = \frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} + \cdots + \frac{\partial^2 f}{\partial x_n^2} \in \mathbb{R}^{1 \times n} \quad (99)$$

67.5 Taylor expansion

테일러 전개(expansion)은 미지의 함수 $f(x)$ 를 $x = a$ 지점에서 근사 다항함수로 표현하는 방법을 말한다. 이는 테일러 급수(series) 또는 테일러 근사(approximation)이라고도 불린다. $f(\cdot)$ 을 $x = a$ 부근에서 테일러 전개를 수행하면 다음과 같이 나타낼 수 있다.

$$f(x)|_{x=a} = f(a) + f'(a)(x - a) + \frac{1}{2!}f''(a)(x - a)^2 + \frac{1}{3!}f'''(a)(x - a)^3 + \cdots \quad (100)$$

함수 $f(\cdot)$ 이 다변수 스칼라 함수일 경우 $\mathbf{x} = \mathbf{a}$ 지점에서 테일러 전개는 다음과 같이 쓸 수 있다.

$$f(\mathbf{x})|_{\mathbf{x}=\mathbf{a}} = f(\mathbf{a}) + \nabla \mathbf{f}(\mathbf{x} - \mathbf{a}) + \frac{1}{2!}(\mathbf{x} - \mathbf{a})^\top \mathbf{H}(\mathbf{x} - \mathbf{a}) + \cdots \quad (101)$$

이 때, $\nabla \mathbf{f}$ 는 함수 $f(\cdot)$ 의 그레디언트(gradients) 의미하며 \mathbf{H} 는 해시안(hessian) 행렬을 의미한다.

68 Reference

1. edwith 인공지능을 위한 선형대수, 주재걸 교수
2. 다크프로그래머 - Gradient, Jacobian 행렬, Hessian 행렬, Laplacian

69 Revision log

- 1st: 2020-05-15
- 2nd: 2020-06-21
- 3rd: 2023-01-21
- 4th: 2023-01-31